

SXC

CODE DEVELOPMENT SYSTEM

The SXC Code Development System provides support for the Scenix Semiconductor SX series of microcontrollers.

Optimizing C language cross compiler is ANSI compatible within the SX hardware limitations.

Tight, fast and efficient code optimization generates clean, customized applications.

Support for the following data types: 8, 16, 24, 32 bit signed and unsigned integers; 16:8, 16:16 fixed point; IEEE-754 (24 and 32 bit) floats; and single bit types.

Includes Standardized Embedded Micro Libraries.

Object libraries can be included directly in C source files.

Includes BCLink linker that runs a final optimization pass after linking.

Built-in Macro Assembler seamlessly incorporates assembly language into C source.

Memory management functions including: named, special (user defined) and local address space.

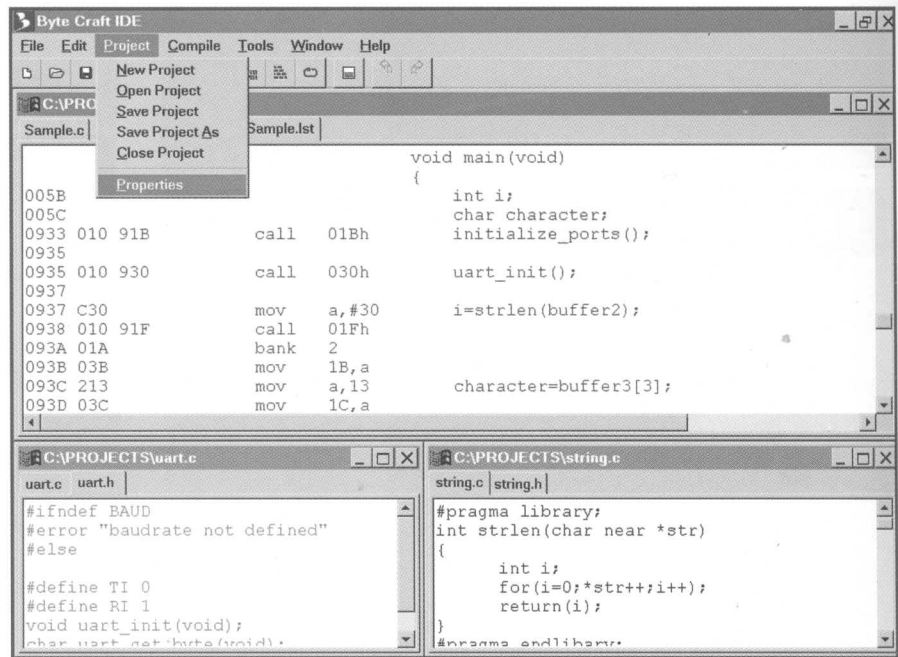
Generates symbol table and source reference information for C source level debugging.

Includes support for interrupt routines.

DOS version includes an editor and an integrated development shell.

Windows 95 version includes BCL Integrated Development Environment for Windows.

The Byte Craft Limited SXC Code Development System is a high-performance package designed for serious developers. The SXC generates small, fast, and efficient code. It enables the professional developer to quickly produce stand-alone single chip microcontroller applications. Developers can easily port C language applications written for other platforms to the SXC.



BCLIDE

The new BCLIDE for Windows 95 provides access to the Byte Craft SXC compiler, Macro-Assembler and BCLink linker. It features project management allowing developers to specify source files, libraries and compiler and linker options, effectively reducing start up time for new projects. Developers can compile individual source files or use the build option to quickly rebuild an entire project. The up-to-date output files assist developers in locating errors and facilitating rapid debugging.

On-line SXC and IDE documentation can be accessed at any phase during the development cycle.

The Byte Craft Integrated Development Environment includes a customizable development toolbar and menu. By adding menu options, users can quickly and conveniently call third party tools. Developers can enhance their SXC Code Development System by adding the FUZZ-C™ preprocessor.

SXC generates full source level debugging information compatible with the Parallax SX Key Development System.



SXC

The **Byte Craft Limited SXC Code Development System** includes one year free technical support to registered owners via telephone, fax, web, BBS or email. Upgrades and maintenance revisions are provided for the first twelve months to registered owners. Complete user documentation is included with every SXC Code Development System.

PRODUCT DEMO

Download a DEMO from our web site
by following the link:
www.bytecraft.com/isxc.html

FEATURES

- BCLink Linker
- Absolute Code Mode
- Fixed Point
- Floating Point
- Named Address Space
- Special Address Space
- Local Address Space
- Stack Space
- Multiple Function Arguments
- Structure Bit Fields
- The bit Data Type
- #pragma Directives
- Integrated Development Environment
- Bit fields can cross byte boundaries

Data Type	Default Size (bits)
bit	1
bits	8
signed/unsigned char	8
signed/unsigned short	8 or 16*
signed/unsigned int	8 or 16*
signed/unsigned int8	8
signed/unsigned int16	16
signed/unsigned int24	24
signed/unsigned int32	32
signed/unsigned long	16 or 32*
float	32
float24	24
float32	32
fixed168	24 bits
fixed1616	32 bits

*set by compiler option



Complex Data Types

Complex data types such as structures, enumerated types, unions, arrays and pointers are supported. Structure bit fields are supported and bit fields can cross a byte boundary. Both near and far pointers and pointers to functions are supported, and pointers to data named address space.

```
struct led{
    int on:1;           // 1 or 0
    int intensity:3;     // 0..7
    int location:5;      // 0..0xf
};
void function1(int i, float f);
int (*function)(int, float)
..
function=function1;
result=(*function)(6,3.141);
```

Support for inline assembly

You can include single and multiple lines of inline assembly within your C program. You can also call an assembly routine from C, call a C function from assembly and pass values between C and assembly routines.

```
int var1;
#ASM
    inc var1
    call func1
#ENDASM
```

Interrupts

The #pragma vector directive specifies the location and assigned name for an interrupt vector.

```
#pragma vector __RTCC @ 0x00;
void __RTCC(void)
{
    ...
}
```

Ports

Ports are declared using the `#pragma port` series of directives. Declare a read only port with `#pragma portr`, a read/write port with `#pragma portrw` and a write only port with `#pragma portw`.

```
#pragma portrw porta @ 0x0090;
...
porta=0x55;
porta.3=1;
```

Devices

The SXC Code Development System is shipped with device specific header files for the SX series of microcontrollers. These header files describe the specific resources of a given device to the SXC compiler. Header files for new devices are made available as soon as devices are released.

Local Variable Space

`#pragma option LOCAL` directive allocates a block of memory for local variable placement. This enables local variable optimization and supports the passing of multiple arguments to functions.

```
0010 0010 #pragma memory LOCAL[0x10]@ 0x10;
                                void func1(void)
                                {
000F 000E 000D 000C            int w,x,y,z;
0005 20E      mov  a,0E        w=x+y+z;
0006 1CD      add  a,0D
0007 1CC      add  a,0C
0008 02F      mov  0F,a
0009 00D      retp
                                }
                                void func2(void)
                                {
000B 000A 0009 0008            int w,x,y,z;
000A 20A      mov  a,0A        w=x+y+z;
000B 1C9      add  a,09
000C 1C8      add  a,08
000D 02B      mov  0B,a
000E 00D      retp
                                }
                                void func3(void)
                                {
000F 000E 000D 000C            int w,x,y,z;
000F 20E      mov  a,0E        w=x+y+z;
0010 1CD      add  a,0D
0011 1CC      add  a,0C
0012 02F      mov  0F,a
0013 010 90A  call 00Ah      func2();
0015 00D      retp
                                }
```

Named Address Space

Named address space allows you to associate a name with a specific area of memory. This is useful when you wish to group variables.

Special Variable Space

Special Variable Space is memory supported by user provided device drivers. Use the `#pragma memory SPECIAL` to access extra RAM.

```
#pragma memory SPECIAL bank2[0x20] @ 0x2ff;
char bank2_r(char loc){
    porta.0=0;
    porta=1;
    return(loc);
}
void bank2_w(char loc,char val){
    porta.0=0;
    porta.1=1;
    loc=val;
}
void main(void){
    int bank2 j;
    j = 0;    ...
}
```

Embedded C Extensions

The SXC compiler supports extensions to the C language which are designed specifically for the embedded systems developer. Some extensions include the bit and bits data types, binary constants, extended case statements, direct variable placement with the `@` symbol, support for processor specific functions and consistency checks or port declaration.

```
000F                                int status;
                                void __STARTUP(void)
                                {
000F                                int var1 @ status;
0053                                int var2 @ 0x53;
0016 2AF  inc  0F        var1++;
0017 01A  bank 2        var2++;
0018 2B3  inc  13
0019 00D  retp          }
```

Libraries

The SXC includes standard ANSI and Standardized Embedded Micro Libraries.

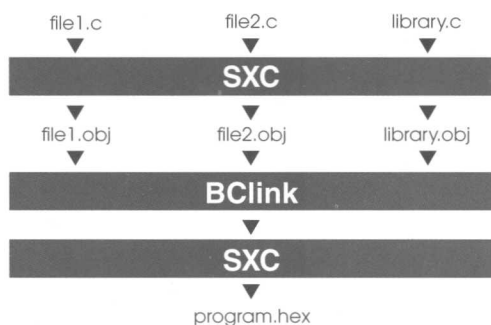
SXC

BLink Linker

Linking in Libraries and Modules using BLink

When you use a linker to link project libraries with other project modules you want to ensure that the most optimized code emerges after the linking process.

Byte Craft's BLink linker links project modules and then sends the generated code to the compiler optimizer to ensure that all new optimization possibilities are found.



Absolute Code Mode

Including Libraries Directly in your Source Code

SXC allows you to include library files in object form in your C source program.

The special directives `#pragma library` and `#pragma endlibrary` are used to create libraries. Then, simply by creating a header file and naming the object library with a `.lib` extension you can place the statement `#include library.lib` in your C language program. Absolute code mode uses an intelligent function inclusion algorithm where only the library functions called are included in the compile.



Can I combine Assembly and C ?

YES, both multiple and single lines of assembly can be included using the `#asm` and `#endasm` directives

Is the SXC ANSI compatible?

YES, within the physical limitations of the SX device. Arrays must fit within 256 bytes, be of one dimension, and can not contain structures or unions.

Are there C extensions for embedded developers?

YES, register variables, bits and bit data types, case statement extensions, direct variable placement, binary constants, processor specific functions and `#pragma` directives.

Can I place read-only data in ROM?

YES, declaring a constant array or placing structures in ROM. A similar technique is used to place basic data types in ROM.



For product information, free demonstration software, pricing or to place an order please contact our office or visit our web site.

Byte Craft Limited Product Line.

C6805; C6808; MPC; COP8C; Z8C; C38; Fuzz-C™; C8051

Byte Craft Limited products are available from our domestic and international distributors or directly from Byte Craft Limited, please contact us for your local distributor.

Byte Craft Limited

421 King Street North
Waterloo, Ontario, Canada N2J 4E4

Phone: (519) 888-6911

Fax: (519) 746-6751

BBS: (519) 888-7626

Email: info@bytecrafft.com

Web: www.bytecrafft.com

The SXC Code Development System runs under MS/PC-DOS or Microsoft Windows 95, HP-UX or Sun Solaris
IBM PC and PC-DOS are registered trademarks of International Business Machines. MS-DOS and Windows are trademarks of Microsoft Corp.